ADDRESS GEOCODING

BACKGROUND OF THE INVENTION

A. Field of the Invention

[0001] The present invention relates generally to networks, and more particularly, to geolocation information associated with resources on a network.

B. Description of Related Art

[0002] The World Wide Web ("web") contains a vast amount of information. Locating a desired portion of the information, however, can be challenging. This problem is compounded because the amount of information on the web and the number of new users inexperienced at web searching are growing rapidly.

[0003] Search engines attempt to return hyperlinks to web documents in which a user is interested. Generally, search engines base their determination of the user's interest on search terms (called a search query) entered by the user. The goal of the search engine is to provide links to high quality, relevant results to the user based on the search query. Typically, the search engine accomplishes this by matching the terms in the search query to a corpus of pre-stored web pages. Web documents that contain the user's search terms are "hits" and are returned to the user.

[0004] Some web documents may be of particular interest to users that reside in certain geographical areas. For example, web documents associated with an on-line newspaper may be of most relevance to the geographical area covered by the newspaper. Web documents associated with local businesses or organizations are additional examples of web documents that may be of particular interest to a geographical area. Thus, it can

be desirable for a search engine to know whether a web document has geographical significance and when it does, the geographical locations associated with the web document.

[0005] Web documents often include postal addresses. In some situations, the postal addresses may help to define the geographical relevance of the web document. More specifically, a postal address can be converted to a geographic coordinate (e.g., latitude and longitude) value. The geographic coordinate can be used to calculate the distance between two locations. In the context of web documents and web searching, a geographically distant web document may be determined to be less relevant than a closer web document.

[0006] Extracting valid addresses from web documents and efficiently geocoding them (i.e., converting the address to geographic coordinate values) can be a difficult problem. Extracting postal addresses can be difficult because addresses can be written in a number of different formats and may not even be complete addresses. The zip code, for example, may be omitted from an address.

[0007] In addition to extracting valid postal address information, accurately and efficiently geocoding the postal addresses can be difficult. Ideally, the geocoding should be able to handle all postal addresses, produce geographic coordinate information that is as close to the actual address as possible, and be able to quickly generate the geocode information.

[0008] Accordingly, there is a need in the art to be able to associate documents with geographic locations by efficiently extracting and geocoding postal addresses.

SUMMARY OF THE INVENTION

[0009] One aspect of the invention is directed to a method for generating geographic coordinate information that includes receiving at least one term that specifies an address and accessing a table defining coordinate information for ranges of addresses to find an intersection of sets of rows in the table that correspond to the at least one term. The method further includes reading geographic coordinate information from the table at the intersection of the sets of rows in the table.

[0010] A second aspect of the invention is directed to a system for geocoding postal addresses. The system includes a table including rows that each correspond to a range of one or more addresses. Each of the rows include fields that define the row. A geocoding component generates geographic coordinate information for a received address specified by one or more terms that correspond to the fields by locating at least one row in the table that corresponds to an intersection of a number of sets of rows defined by the terms in the received address.

[0011] Yet another aspect of the invention is directed to a method for extracting addresses from a document. The method includes identifying possible address terms based on predetermined rules, verifying that the identified possible address terms are address terms by comparing the address terms to a table containing known addresses, and examining a relative position of the verified possible address terms in the document to determine whether the verified possible address terms form a valid address.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, explain the invention. In the drawings,

[0013] Fig. 1 is an exemplary diagram of a network in which systems and methods consistent with the principles of the invention may be implemented;

[0014] Fig. 2 is an exemplary diagram of a client or server device according to an implementation consistent with the principles of the invention;

[0015] Fig. 3 is an exemplary functional block diagram illustrating an implementation of the server software shown in Fig. 1;

[0016] Fig. 4 is a diagram illustrating an exemplary implementation of a table used by the geocoding component shown in Fig. 3;

[0017] Fig. 5 is a flow chart illustrating operations consistent with the present invention for geocoding addresses; and

[0018] Fig. 6 is a flow chart illustrating exemplary operations for extracting addresses from text documents.

DETAILED DESCRIPTION

[0019] The following detailed description of the invention refers to the accompanying drawings. The detailed description does not limit the invention.

[0020] As described herein, a geocoding component includes a table of sorted address fields that allow the geocoding component to efficiently associate postal addresses with address coordinate values (e.g., latitude and longitude values). The geocoding

component may receive the postal addresses to geocode from another component or the geocoding component may itself extract the postal addresses from text documents.

EXEMPLARY NETWORK OVERVIEW

[0021] Fig. 1 is an exemplary diagram of a network 100 in which systems and methods consistent with the principles of the invention may be implemented. Network 100 may include multiple clients 110 connected to one or more servers 120 via a network 140. Network 140 may include a local area network (LAN), a wide area network (WAN), a telephone network, such as the Public Switched Telephone Network (PSTN), an intranet, the Internet, or a combination of networks. Two clients 110 and a server 120 have been illustrated as connected to network 140 for simplicity. In practice, there may be more or fewer clients and servers. Also, in some instances, a client may perform the functions of a server and a server may perform the functions of a client.

[0022] Clients 110 may include client entities. An entity may be defined as a device, such as a wireless telephone, a personal computer, a personal digital assistant (PDA), a lap top, or another type of computation or communication device, a thread or process running on one of these devices, and/or an object executable by one of these device. Server 120 may include server entities that process, search, and/or maintain documents in a manner consistent with the principles of the invention. Clients 110 and server 120 may connect to network 140 via wired, wireless, or optical connections.

[0023] Clients 110 may include client software such as browser software 115. Browser software 115 may include a web browser, such as the Microsoft Internet

Explorer or Netscape Navigator browser. For example, when network 140 is the Internet, clients 110 may navigate the web via browsers 115.

[0024] Server 120 may operate as a web server and include appropriate web server software 125. In one implementation, web server software 125 may function as a search engine, such as a query-based web page search engine. In general, in response to client requests, search engine 125 may return sets of documents to clients 110. The documents may be returned to clients 110 as a web page containing a list of links to web pages that are relevant to the search query. This list of links may be ranked and displayed in an order based on the search engine's determination of relevance to the search query. Although server 120 is illustrated as a single entity, in practice, server 120 may be implemented as a number of server devices.

[0025] A document, as the term is used herein, is to be broadly interpreted to include any machine-readable and machine-storable data. A document may be an e-mail, a file, a combination of files, one or more files with embedded links to other files, a news group posting, a web advertisement, etc. In the context of the Internet, a common document is a Web page. Web pages often include content and may include embedded information (such as meta information, hyperlinks, etc.) and/or embedded instructions (such as Javascript, etc.).

EXEMPLARY CLIENT/SERVER ARCHITECTURE

[0026] Fig. 2 is an exemplary diagram of a client 110 or server 120 according to an implementation consistent with the principles of the invention. Client/server 110/120 may include a bus 210, a processor 220, a main memory 230, a read only memory

(ROM) 240, a storage device 250, one or more input devices 260, one or more output devices 270, and a communication interface 280. Bus 210 may include one or more conductors that permit communication among the components of client/server 110/120.

[0027] Processor 220 may include any type of conventional processor or microprocessor that interprets and executes instructions. Main memory 230 may include a random access memory (RAM) or another type of dynamic storage device that stores information and instructions for execution by processor 220. ROM 240 may include a conventional ROM device or another type of static storage device that stores static information and instructions for use by processor 220. Storage device 250 may include a magnetic and/or optical recording medium and its corresponding drive.

[0028] Input device(s) 260 may include one or more conventional mechanisms that permit a user to input information to client/server 110/120, such as a keyboard, a mouse, a pen, voice recognition and/or biometric mechanisms, etc. Output device(s) 270 may include one or more conventional mechanisms that output information to the user, including a display, a printer, a speaker, etc. Communication interface 280 may include any transceiver-like mechanism that enables client 110 to communicate with other devices and/or systems. For example, communication interface 280 may include mechanisms for communicating with another device or system via a network, such as network 140.

[0029] The software instructions defining server software 125 and browser software 115 may be read into memory 230 from another computer-readable medium, such as data storage device 250, or from another device via communication interface 280. The software instructions contained in memory 230 causes processor 220 to perform

processes that will be described later. Alternatively, hardwired circuitry may be used in place of or in combination with software instructions to implement processes consistent with the present invention. Thus, implementations consistent with the principles of the invention are not limited to any specific combination of hardware circuitry and software.

[0030] As mentioned, server software 125 may implement a search engine that, based on a user query, returns a list of links to documents that the server software considers to be relevant to the search. Server software 125 may return additional information to the user, such as web advertisements and descriptive information summarizing the documents in the list of links. Consistent with aspects of the invention, server software 125 may perform geocoding functions.

SERVER SOFTWARE 125

[0031] Fig. 3 is an exemplary functional block diagram illustrating an implementation of server software 125. Server software 125 may include a search component 305, a database component 310, and a geocoding component 315. In general, search component 305 may receive user search queries from clients 110, search database 310 based on the search queries, and return lists of links (e.g., URLs) of relevant documents to clients 110. A list of links may also include information that generally attempts to describe the contents of the web documents associated with the links. The list of links may be ordered based on ranking values that are generated by search component 305 and that rate the links based on relevance.

[0032] Database component 310 may store an index of web documents. The web documents may have been previously downloaded and stored in database component 310

from resources on network 140. Database component 310 may be updated as new web documents are downloaded and added to database component 310. Database component 310 may be accessed by search component 305 when responding to user search queries.

[0033] Geocoding component 315 performs geocoding functions in a manner consistent with aspects of the invention. In one implementation, geocoding component 315 may receive postal addresses from search component 305 and geocode the postal addresses to return geographic coordinate information to search component 305. In other implementations, geocoding component 315 may receive documents, such as web documents, extract postal addresses from the received documents, and geocode the extracted postal addresses.

[0034] Although geocoding component 315 is shown in conjunction with search component 305 and database component 310, geocoding component 315 may be implemented independently of these two components. Thus, geocoding component 315 could be implemented as a stand alone geocoding component. Alternatively, geocoding component 315 could be implemented in conjunction with other components, such as an advertisement component that is designed to return geo-relevant advertisements.

OPERATION OF GEOCODING COMPONENT

[0035] Geocoding component 315 may perform certain of its geocoding functions using a table relating to postal addresses, such as all U.S. postal addresses. Fig. 4 is a diagram illustrating an exemplary implementation of such a table, labeled as table 400.

[0036] Table 400 may include a number of rows 401. Each row 401 may correspond to an address or a range of addresses. Rows 401 may each include a number of field

(column) values that define the address(es) associated with a particular row. As shown, the field values may be defined as state field 410, county field 411, city field 412, zip code field 413, street base name field 414, street field 415, parity value field 416, starting street number field 417, ending street number field 418, starting latitude value field 419, starting longitude value field 420, ending latitude value field 421, and ending longitude value field 422.

[0037] State field 410, county field 411, city field 412, and zip field 413 describe the state, county, city, and zip code with which the address(es) defined by the row is associated. Street base name field 414 may define the base portion of the street with which the row is associated. The base portion may include just the "unique" portion of the street name and not include common street descriptive information such as road, street, parkway, etc. Street field 415 may include the complete street name. Parity field 416 may be used to store whether address(es) in the particular row includes just odd or even numbers. For many blocks of street addresses, buildings are numbered consecutively but use only odd or even numbers. Parity field 406 indicates whether the street numbering system uses odd or even numbers. A parity value of "0", for example, might represent even numbers and a value of "1" might represent odd numbers.

[0038] Each row 401 may define more than a single address. Starting street number field 417 lists the beginning (lower) street number in the range covered by a particular row 401. Ending street number field 418 lists the ending (highest) street number in the range covered by a particular row 401. Starting latitude field 419 and starting longitude field 420 together define a geographic coordinate for the address defined by starting street number field 417 of the row. Similarly, ending latitude field 421 and ending

longitude field 422 together define a geographic coordinate for the address defined by ending street number field 418 of the row.

[0039] Exemplary values for one of rows 401 are shown in Fig. 4. This row includes the values: CA (field 410), Santa Clara (field 411), Mountain View (field 412), 94043 (field 413), Bayshore (field 414), Bayshore Pkwy (field 415), 0 (even parity) (field 416), 2100 (field 417), 2640 (field 418), 37.422710 (field 419), -122.094522 (field 420), 37.427439 (field 421), and -122.099373 (field 422).

[0040] Table 400 may include rows that may collectively cover addresses for a large geographical area, such as all postal addresses in the United States. Accordingly, table 400 may be a relatively large table.

[0041] Fig. 5 is a flow chart illustrating operations consistent with the present invention for geocoding addresses. Geocoding component 315 may begin by receiving a postal address to geocode (act 501). The address may be specified as a number of terms that correspond to certain ones of the fields in table 400 (e.g., state, county, city, zip code, street name, street number). Each term in the address corresponds to a subset of the rows of table 400. The row corresponding to the complete address can be found as the intersection of each specified subset (act 502).As an example of finding the intersection of a number of address fields, consider the address "2400 Bayshore Pkwy, Mountain View, CA". Geographic component 315 may first determine the three sets of rows: (1) the set of rows with street (field 415) equal to "Bayshore Pkwy," (2) the set of rows with city (field 412) equal to "Mountain View," and (3) the set of rows with state (field 410) equal to "CA." Geocoding component 315 then intersects these three sets of rows, with the resultant rows corresponding to "Bayshore Pkwy, Moutain View, CA." From within

these resultant rows, geocoding component 315 may find the one row containing street number 2400. The intersection operation used to find the intersection of the specified address fields can be performed efficiently. In one implementation, table 400 is stored such that it is sorted by columns 410-418 in the sorting order: state field 410, county field 411, city field 412, zip code field 413, street base name field 414, street field 415, parity field 416, starting street number field 417, and ending street number field 418, where state field 410 is the primary sorting key, county field 411 is the secondary sorting key, and so on. By sorting in this manner, each set of rows that are used to perform the intersection can be specified by a relatively few "blocks" of rows.

[0042] In the example of "2400 Bayshore Pkwy, Mountain View, CA", the set of rows with state name "CA" in field 410 is a single block, since table 400 is primarily sorted by field 410. The set of rows with city name "Mountain View" in field 412 can be specified by multiple blocks (such as a few dozen) blocks, with one block per each of the cities in the U.S. with the name "Mountain View." Since each set is specified by a relatively few "blocks" of rows, the intersection of sets may be performed very efficiently.

[0043] Implementations consistent with principles of the invention can also reduce the size of table 400. For example, instead of filling in column 401 (state name) for every row in table 400, geocoding component 315 can just record for each state name its starting row number and ending row number.

[0044] Geocoding component 315 may operate to geocode incompletely specified addresses. Again, consider the example of "2400 Bayshore Pkwy, Mountain View, CA." Although there are multiple cities in CA with the name "Mountain View," only the

Mountain View city in Santa Clara County has a parkway with the name "Bayshore Pkwy." Thus, geocoding component 315 is still able to efficiently geocode this address using intersections of the sets of rows specified by the terms "Bayshore Pkwy," "Mountain View," and "CA."

[0045] After locating a row that corresponds to the received address, geocoding component 315 may read the latitude and longitude information from the located row (act 503). In particular, the starting latitude and starting longitude values may be read from fields 419 and 420. The ending latitude and ending longitude values may be read from fields 421 and 422.

[0046] In some situations, geocoding component 315 may also receive a particular street number with the postal address, for example, "2400" in the address "2400 Bayshore Pkwy, Mountain View, CA.." For example, in the address given above, the street number may be specified as 2400 Bayshore Parkway. Using the street number, the starting street number specified in field 417, the ending street number specified in field 418, and the latitude and longitude pairs read in act 503, geocoding component 315 may interpolate the starting and ending latitude and longitude pairs to find a latitude and longitude pair that corresponds to the received address (act 504). In one implementation, the interpolation may be a linear interpolation. If the received address does not contain a particular street number, geocoding component 315 may simply use, for example, an average of the starting and ending latitude/longitude pairs as the output latitude/longitude pair.

[0047] In the implementations discussed above, geocoding component 315 receives a postal address to geocode (act 501). This address may be acquired in a number of ways,

such as extracting the addresses from documents or receiving them from address fields of online or offline forms. In one exemplary implementation, the addresses may be extracted from text, such as text from web pages. The addresses may be extracted by geocoding component 315, search component 305, or other components.

[0048] Fig. 6 is a flow chart illustrating exemplary operations for extracting addresses from text documents. To begin, a text document may be analyzed to locate strings (possible address terms) that refer to possible location names, zip codes, or phone numbers (act 601). These strings may be identified based on a number of predetermined address rules. One set of such rules may be a comparison of the strings in the document to a list of possible address terms. For example, a county name should be followed by the word county. Street names are generally identified by terms such as "street," "road," "drive," "parkway," "pkwy," etc. In addition to a straight textual matching to determine whether a string is an address term, geocoding component 315 may look for capitalization that is consistent with a written address. For example, geocoding component 315 may require that street and city names be capitalized. Still further, street names may be required to be preceded by a number, e.g., "2500 Bayshore Pkwy." Zip codes may be identified as five-digit strings and potential phone numbers may be identified as three digits followed by seven digits or three digits, followed by three digits, followed by four digits.

[0049] The initial set of strings determined in Act 601 may be normalized to standardize the address terms (act 602). A table of abbreviations, such as the United States Postal Service (USPS) table of street abbreviations may be used to normalize the terms. In the normalization process, abbreviations are converted to a single

representation. For example, "Sreet." may be converted to "St" and "Parkway" and "Pky" are converted to "Pkwy."

[0050] Geocoding component 315 may lookup the normalized address terms in table 400 (act 603). Address fields that are not in table 400 may be discarded. In some situations, street prefix/suffixes are occasionally either wrong or omitted in written addresses. To deal with these addresses, geocoding component 315 may only lookup the base name of streets in table 400. Thus, the street name "Bayshore Pkwy" may be looked-up as "Bayshore."

[0051] Once candidate address terms are identified in Acts 601-603, heuristics may be applied to combine candidate addressterms into true addresses, which may then be verified by looking up in table 400 (act 604). In general, the heuristics may relate to the relative positioning of the address terms. For example, a city name should be followed by a state or a zip code, such as "Mountain View, CA" or "Mountain View, 94043." As another example, a street name should be followed, within a certain number of characters, by a city, state, or zip code term. This allows for terms such as floor or suite numbers to be between the street name and the city, state, or zip code. For example, the address:

1200 Abernathy Road
600 Northpark Town Center
Suite 1700
Atlanta, GA 30328

includes two lines of address terms between the street number and the state. In general, one of ordinary skill in the art will recognize that other heuristically-generated rules could be applied to determine if a set of address terms define a valid address.

[0052] The addresses located by geocoding component 315 in the text documents are particularly suited to being converted to geographic coordinate information using the above-described techniques as these techniques can operate on partial addresses and are efficient.

[0053] In some situations, such as in web directory listings, a number of addresses may be listed for a certain city or other geographic area. For example, a restaurant guide may list a number of restaurants in a requested city. Although the requested city may be listed at the top of the restaurant guide (e.g., "Restaurant Listings For Mountain View"), each particular restaurant address may not explicitly include the city, as it is assumed that the reader will know that all the restaurants are in the same city (e.g., Mountain View). In these situations, geocoding component 315 may associate each address with the city listed near the top of the web page. Additionally, phone numbers given for each of the restaurants may be used to further infer the correct city for the restaurants.

CONCLUSION

[0054] Geocoding component 315, as described above, may perform a number of geographic relevance related functions, such as extracting postal addresses from text documents and geocoding postal addresses. A table may be used by geocoding component 315 when geocoding addresses. The table includes a number of rows, each corresponding to one or more addresses. Geocoding component 315 can quickly locate a particular row based on a number of input address fields as the intersection of the sets of rows that correspond to each of the address fields.

[0055] It will be apparent to one of ordinary skill in the art that aspects of the invention, as described above, may be implemented in many different forms of software, firmware, and hardware in the implementations illustrated in the figures. The actual software code or specialized control hardware used to implement aspects consistent with the present invention is not limiting of the present invention. Thus, the operation and behavior of the aspects were described without reference to the specific software code--it being understood that a person of ordinary skill in the art would be able to design software and control hardware to implement the aspects based on the description herein.

[0056] The foregoing description of preferred embodiments of the present invention provides illustration and description, but is not intended to be exhaustive or to limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practice of the invention. For example, certain of the acts performed in Figs. 5 and 6 may be performed in parallel or in a different order.

[0057] No element, act, or instruction used in the description of the present application should be construed as critical or essential to the invention unless explicitly described as such. Also, as used herein, the article "a" is intended to include one or more items. Where only one item is intended, the term "one" or similar language is used.